パートナー様 導入事例：001

株式会社サイシード：コロナワクチン接種専用予約管理システム

特殊な要件の多いワクチン接種予約のシステムを、AWSサービスをフル活用することによって速やかに開発。

『コロナワクチン接種専用予約管理システム』について

「コロナワクチン接種専用予約管理システム」は、新型コロナウイルスのワクチン接種の予約及び予約管理を行うためのシステムです。住民は、web ブラウザ・LINE・電話（AmazonConnect を使用した音声 IVR システム）を使用して予約することができ、それぞれの予約は、自治体（事務局）側で一元管理することが可能となります。コールセンターのオペレーターが住民から電話を受けて、代わりに予約することも可能。さらに、住民からの問い合わせに自動で対応する、AI チャットボットも標準プランで搭載されています。

開発要件・課題

通常の予約システムと大きく異なる、コロナワクチン接種予約システムの開発要件

●2 回目の接種は、1 回目から規定の期間をあけないと接種予約ができない。
●その規定の期間は、ワクチンの種類によって異なる。
●1 回目と 2 回目で、異なるワクチンを予約できない。
●予診のみの人は、接種済みとしてカウントしない。
●コールセンターと個別接種会場では、行える操作が違う。
●アクセスがかなり集中した際でもシステム停止しないよう、様々な負荷対策が必要。

これらのルールをシステムに組み込むことで、現場のオペレーションの最適化を行いました。サービスの構築の課題としては、

●数万人から数十万人の利用者の情報を扱えるような、スケーラビリティ性が求められる。
●予約が集中しシステムダウンする可能性があり、冗長性が必要。
●電話の予約も対応ができる。
●上記を満たした上で、十分なセキュリティを担保する。

このような要件や課題を、ごく僅かな期間と限られた人的リソースでクリアする必要がありました。

数百万単位のアクセス集中に耐えうるサービスとして、半年で構築に成功。

サービスが停止してしまうようなトラブルもなく、安心していただけるサービスとして、2021 年 11 月現在、200 以上の自治体（対象人口 3000 万人）が導入。人口 100 万人を超える大規模な自治体から、1 万人程度の小規模な自治体まで幅広く利用しています。

AWS のメリット / AWS だからこそクリアできたポイント

●IVR も利用したシステムの構築ができる。
●スケーラビリティに優れたサービスが多く（特に Aurora RDS）、かんたんに増強などができる。
●利用者たちの知見も多く、提供されているサービスを試す前に事前に情報を得ることができる。
●CodeBuild, CodeDeploy といった、開発者たちの負担が減るようなサービスのフローが構築されている。
●より運用者が安心して監視できるような Cloudwatch といった監視サービスを提供している。
●セキュリティ面でも、システムの監視や保護を増強してくれるサービスが充実している。
●Amazon Connect のように、電話を使った Web 以外のサービスにも対応できる。
●ランニングコスト面でのスケーラビリティ性があるため、パフォーマンスや規模に応じて調整できる。
●サーバなどの価格は公開されているため、月々のランニングコストの計算は規模に対してある程度見積もることができる。
●リザーブドインスタンスのような、大きくランニングコストを引き下げることができるような使い方も選べる。

システム情報

1. フロントエンド

●管理画面とエンドユーザの画面を分けることで、それぞれ提供するサービスを明確化。
●作業に特化した SPA を配信する構成に。
●CodeBuild を利用し、デプロイまでのフローが属人化しないように工夫。

2. バックエンド

●Public, DMZ, Private でオンプレの経験を活かしやすいシンプルな構成。
●アクセス関連のセキュリティが担保しやすい構成。
●開発者が極力サーバへ入ることがないように、CodeDeploy や SystemManager などを通じて、確実な痕跡が残せるような構成。

アクセスが集中しても耐えうる構成にするため、
・CloudFront でキャッシュできるものはキャッシュで配信
・WAF でのレートブロックなどを実施して、悪意のある攻撃を防ぐ
・万が一さばききれない場合は、Lambda の方へリクエストを流して、アクセス集中を説明するレスポンスを返す

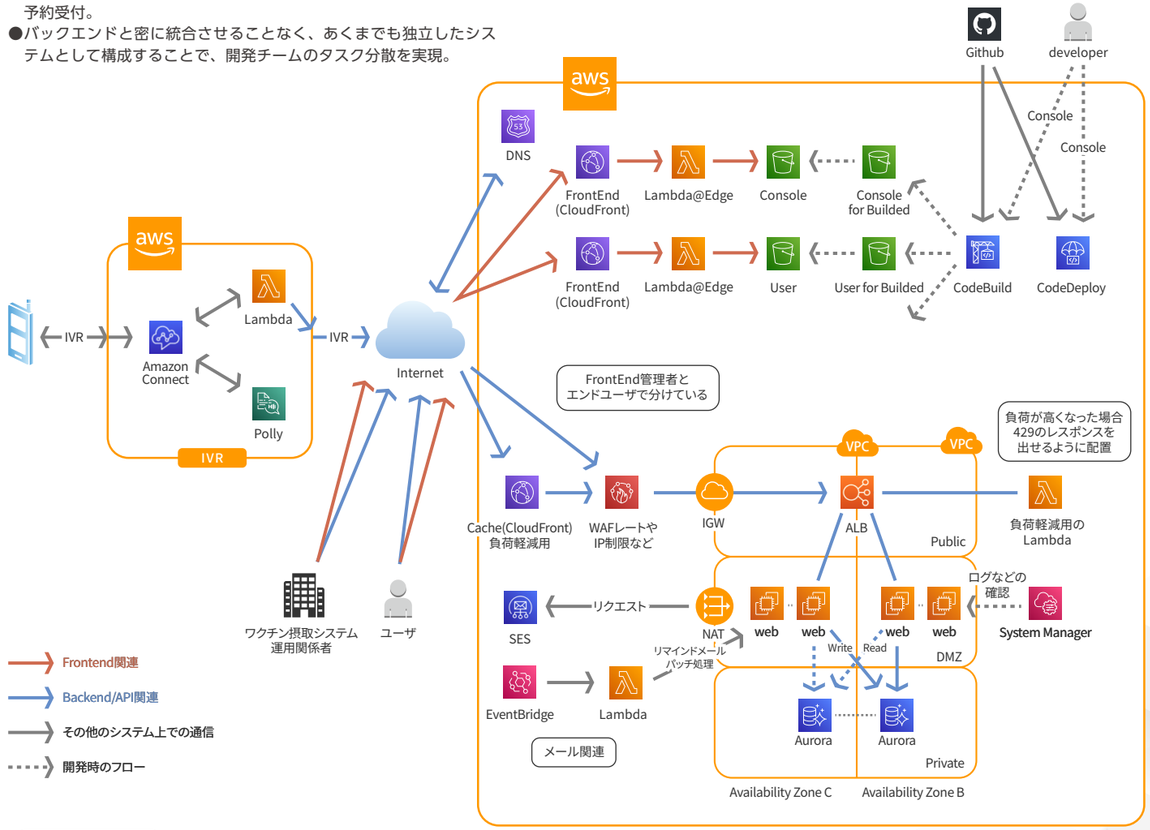
DB も、予め Aurora RDS を利用して、スケーラビリティを確保した形で運用

3. IVR

●主に Connect を利用し、合成音声や電話機からの入力を行っての予約受付。
●バックエンドと密に統合させることなく、あくまでも独立したシステムとして構成することで、開発チームのタスク分散を実現。

<導入した AWS のサービスの構成>

・VPC: バックエンドのネットワーク構築
・WAF: バックエンドのアクセス制御
・ALB: ロードバランサー
・EC2: メインサーバ
・KMS: ストレージの暗号化
・RDS: 予約枠などの管理
・S3: ログの保管など
・Route53: ドメインの管理
・CloudFront: フロントエンド、キャッシュの配信
・SES: メール配信
・GuardDuty:AWS 内における変更の脅威に対する追跡
・Shield:DDos 対策
・CloudWatch: システム監視
・CodeBuild: フロントエンドのビルド用
・SNS:Cloudwatch の通知用
・chatbot:Cloudwatch の通知
・AmazonConnect:IVR による自動新規予約機能



今後の展望

従来のようなオンプレミスに近い構成で経験を生かした構築できることと、よりモダンであるサーバレスアーキテクチャを混合させるような構成を取れることが AWS を利用する上でのメリットだと思っています。今回はサーバのレスポンスが、ユーザ体験や運用にかなり影響を与える側面があったため、パフォーマンスを読みやすいオンプレミスよりの構成を採用していますが、今後はサーバレスアーキテクチャをもっと活用したシステムの構築にシフトしていきたいと考えています。

株式会社サイシード お問い合わせ　TEL：03-6871-8691 Mail：info@sciseed.j